



## Organizando el código

### Transcripción

[00:00] Hola. En el video anterior, nosotros creamos la clase `RegistroDeProducto` que fue la clase donde instanciamos nuestro primer producto, configuramos sus atributos y luego instanciamos el `EntityManager` que es el que nos permite manejar las transacciones y los estados para hacer persistido en la base de datos.

[00:19] Adicionalmente creamos dos propiedades en el archivo `peresistence.xml`, una propiedad que nos permitía ver la operación que estaba siendo realizada en la base de datos y la otra propiedad creaba las tablas y las columnas de forma automática por nosotros.

[00:34] Ahora, nosotros vamos a organizar un poco este código creando una clase DAO para la entidad producto, recordando que el DAO simplemente tiene la responsabilidad de dar acceso a la base de datos, o sea para guardar, actualizar, eliminar o para consulta de una entidad.

[00:54] Y vamos a crear una otra clase que es una clase utilitaria para asignar la responsabilidad de creación del `EntityManager`. Vamos a comenzar. En la carpeta `tienda` vamos a crear una nueva clase. Esa clase se va a llamar `producto DAO`, va a estar dentro del paquete `DAO`. Damos finalizar.

[01:22] Entonces esta clase va a tener los métodos de consulta, así como de guardar o actualizar un registro. Para eso tiene que utilizar el `EntityManager`, por lo que se lo vamos a pasar como atributo hacer un atributo privado del tipo `EntityManager`.

[01:44] Lo vamos a llamar em, tenemos que importar el EntityManager y lo vamos a pasarlo como parámetro en el constructor. Entonces vamos en el constructor con parámetro EntityManager y ahora sí podemos comenzar a construir nuestros métodos de acceso.

[02:04] El primer método va a ser el método guardar. Es un método público que no retorna nada, guardar, recibe una entidad del tipo producto. El nombre va a ser producto y vamos a utilizar el método persist del entityManager. Nos está solicitando una entidad que va a ser el producto.

[02:33] Ya con eso tenemos nuestro primer método creado, está todo funcionando correctamente, importamos la clase producto y vamos a ir a nuestra clase main donde vamos a instanciar ese Dao para poder reemplazar el método de persistencia.

[02:58] Vamos a llamar productoDao, usamos la palabra clave new, ProductoDao y nos va a solicitar primero que importemos el producto y segundo que pasemos un parámetro que va a ser el EntityManager.

[03:16] Con eso, nosotros podríamos enviar la parte de iniciar la transacción y de cerrar la transacción así como el commit, podríamos colocarla dentro del método guardar, pero ahí tendríamos que duplicar el código en los diferentes métodos.

[03:34] Entonces, para evitar duplicación de código, simplemente vamos a iniciar las transacciones, vamos a realizar las diferentes operaciones. La primera operación que vamos a usar acá es la de guardar productoDao, vamos a llamar el método guardar que lo acabamos de crear y vamos a enviar el producto que va a ser el celular.

[03:56] Iniciamos la transacción. Realizamos la persistencia, lo enviamos ese valor a la base de datos y cerramos la conexión. Lo siguiente sin asignar la responsabilidad de crear el EntityManager, para eso vamos a crear otra clase.

Vamos a ir a nuevo, clase. Esa clase se va a llamar JPAUtils y va a estar dentro del paquete utils.

[04:30] Vamos a colocarlo en el utils. Presionamos finalizar. Lo primero que tenemos que hacer es crear el factory, con el createEntityManagerFactory, que es el atributo que nos va a permitir crear el EntityManager. Vamos a crear un atributo estático. Va a ser del tipo EntityManagerFactory. Lo vamos a llamar FACTORY.

[05:05] Tenemos que importar esa clase, y recordando lo que habíamos visto en la clase anterior, esto proviene de la clase persistence, un método estático que sería createEntityManagerFactory, nos va a solicitar el nombre de esa base de datos que nosotros la llamamos tienda.

[05:28] Recordando en la clase del archivo persistence, nosotros habíamos colocado el nombre para esta base de datos tienda, que está aquí. Si ustedes le colocaran otro nombre, tiene que ser el nombre que le asignaron a la unidad de persistencia.

[05:46] Entonces de vuelta en la clase utilitaria, ya tenemos el Factory, que nos va a permitir crear el EntityManagerFactory. Nosotros ahora vamos a quedar un método, que va a ser público y va a retornar un EntityManager, ese lo vamos a llamar getEntityManager.

[06:10] Entonces, con ese método, ese método también va a ser estático, vamos a importar la clase EntityManager, vamos aquí "Ctrl + V", importamos de aquí cometí un error. EntityManager. Ahora sí.

[06:44] Ahora tenemos que retornar el valor. Va a ser return FACTORY y va a ser create EntityManager. Ya con eso, aquí importamos. Está todo creado. Ahora sí tenemos nuestro método estático creado que nos va a permitir crear el EntityManager.

[07:13] Volvemos a la clase main, vamos a eliminar esta línea de acá y esta de acá la vamos a reemplazar por el método estático que creamos de la clase JPAUtils.getEntityManager.

[07:31] Entonces, ya con eso, nosotros hemos simplificado nuestro código un poco más, un poco más elegante y creamos la clase Dao que es donde vamos a tener todos los métodos de consulta. En la siguiente clase, nosotros vamos a hablar un poco sobre los relacionamientos entre la clase producto y las categorías de ese producto.